LA-UR-19-30301

Approved for public release; distribution is unlimited.

Title: Phase-aware Equation of State Lookups: Addendum

Author(s): Sheppard, Daniel Glen
Pimentel, David A.
Sjostrom, Travis
Crockett, Scott

Intended for: Report

Issued: 2019-10-10

Phase-aware Equation of State Lookups: Addendum
October 9th 2019
Sheppard, Daniel G.; Pimentel, David A.; Sjostrom, Travis; and Crockett, Scott.

This is a short, written addendum to the talk given on the progress and status of the phase aware multiphase project LA-UR 09-30168.

The equation of state (EOS) project is working toward treating the material representation of phases, solids and liquids, as separate tables in an effort have more control of the physical material representations as required data for hydrocodes. In this context, table is defined to be tabulated values of a thermodynamic state variable, such as pressure or energy, on a density and temperature grid. The current multiphase representation of the data is a "flattened" table where each individual phase is tabulated according to thermodynamic stability. The EOS table(s) represent thermodynamic equilibrium such that the derivative of the pressure with respect to density, $\frac{\partial P}{\partial \rho}|_T$, in the mixed phase region is zero along a given isotherm. Whereas, $\frac{\partial P}{\partial \rho}|_T > 0$ in a pure phase. Derivative discontinuities in energy also exist at phase boundaries. The consequence of flattening these phase boundaries into a single table is that higher order interpolators, which are well behaved in pure phases, can give non-physical numerical behavior near phase transitions. In addition, representation of the transition through boundary is not smooth due to grid effects, and the boundaries tend to exhibit a saw tooth pattern along the grid. These types of data insufficiencies can cause unexpected behavior in hydrocodes and often have to be dealt with via complicated coding exceptions on the host code side. The goal of this work is to provide improved data constructs that eliminate these types of errors, lets the host code use the data as more of a black box, and better represent the actual physics of equilibrium phase changes. Another benefit will be a data frame work upon which more complicated models can be easily built, such as a kinetic phase change model or phase information for a phase-aware strength model.

At a high level, a prototype for utilization of this kind of data requires the following: a code/solver to produce the phase boundary data, a standardized way to store this data in our EOS library frame work called SESAME, an adaptation of the SESAME library to store individual phases, new algorithms to identify upon which table to operate given variable inputs, and new algorithms that use the phase boundary data to calculate mixed phase quantities. At this point we have a working prototype code, written in python, that solves for the phase boundaries using a minimization of Gibbs free energy. We have an nearly-finalized file format for the new SESAME format. We also have a prototype Perl implementation, which utilizes the phase boundary data to determine the appropriate location in phase-space and relevant phase(s). The Perl prototype currently uses either pairs of density and energy or pairs of density and temperature as input. To implement this for pairs of pressure and temperature, some physics-based assumptions and decisions will be required since (P,T) is insufficient for calculating the phase fractions of mixed phase regions. As a next step, we need to write an interpolation algorithm for mixed phases and triple-points, and we will need to perform general robustness testing.

To achieve a production ready software implementation, we need to generalize the python boundary solver and implement it in our EOS production software OpenSesame. We also require a final decision on the SESAME format requirements to produce the tables in a readable format. We need to adapt our library input/output software, SESIO, to read and write the new tables. The Perl prototype will also need to be implemented in EOSPAC; which will likely require some public API changes so that the host codes can utilize the new features (i.e., queries about the current phase). A full production release is like a couple years away; however, in the next fiscal year, we are planning a limited alpha-release of EOSPAC, which will incorporate use of phase boundaries.